

09-20-00

A

UNITED STATES PATENT APPLICATION TRANSMITTAL FORM

BOX PATENT APPLICATION  
ASSISTANT COMMISSIONER FOR PATENTS  
Washington, D.C. 20231

Docket No.: 909.0028USU

Sir:

Transmitted herewith for filing is the patent application of

Inventor(s): Jean-Francois Le Pennec; Olivier Hericourt and Jacques Fieschi

For: METHOD AND SYSTEM FOR GENERATING AND USING A  
VIRUS FREE FILE CERTIFICATE

Enclosed are:

XXX Specification (24 pps.) consisting of: Description (19 pps); Claims (4 pps);  
Abstract (1pp);

XXX Declaration and Power of Attorney;

XXX Appointment of Associate Power of Attorney;

XXX 5 sheets of drawing;

XXX An assignment of the invention to: International Business Machines Corporation, including \$40.00 recordation fee and Assignment  
Recordation Form Cover Sheet;

\_\_\_\_ Verified Statement (Declaration Claiming Small Entity Status - Small  
Business Concern;

\_\_\_\_ Information Disclosure Statement (with copies of patent);

\_\_\_\_ Form - PTO-1449;

XXX Priority of application Serial No. 00480006.6 filed on 06 January 2000,  
in Europe is claimed under 35 U.S.C. §119;

XXX Priority Document of application Serial No. 00480006.6 filed on January  
6, 2000 in Europe; and

09/19/00

10894 U.S. P.

10894 U.S. PTO

09/665040

09/19/00

09/665040: 09/19/00

**XXX** Preliminary Amendment.

The Filing Fee is calculated below.


CLAIMS AS FILED				
(1) For	(2) Number Filed	(3) Number Extra	(4) Rate	(5) Basic Fee \$690.00
Total Claims	17 - 20 =	0	x \$18.00	\$0
Independent Claims	5 - 3 =	2	x \$78.00	\$156.00
Multiple Dependent Claim Fee		x \$260.00 = \$0.00		
<b>TOTAL FILING FEE</b>		<b>\$846.00</b>		
1/2 FILING FEE FOR SMALL ENTITY		\$		

**XXX** A check in the amount of \$ **886.00** to cover the filing fee (\$846) and assignment recordal fee (\$40) is enclosed.

**XXX** The Commissioner is hereby authorized to charge any additional fees under 37 C.F.R. §§1.16 and 1.17 which may be required with this communication or during the entire pendency of the application, or credit any overpayment, to **Deposit Account No. 01-0467**. A duplicate copy of this Form is enclosed.

Address all future communications to: **Harry F. Smith, Esq.**  
**Ohlandt, Greeley, Ruggiero & Perle, L.L.P.**  
**One Landmark Square, 10th Floor**  
**Stamford, Connecticut 06901-2682**  
**U.S.A.**

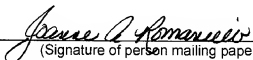
September 19, 2000  
Date of Signature

  
Harry F. Smith, Esq.  
Attorney for Applicant(s)  
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.  
Registration No. 32,493  
Telephone: (203) 327-4500  
Telefax: (203) 327-6401

**CERTIFICATE OF EXPRESS MAILING**

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Certificate No. **EL562862914US**, service under 37 CFR §1.10 and is addressed to: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231 on September 19, 2000.

Joanne A. Romaniello  
(Typed name of person mailing paper)

  
(Signature of person mailing paper)

EXPRESS MAIL NO. EL 562 862 914 US

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent Application of:

INVENTORS: Jean-François LE PENNEC, Olivier HERICOURT, and Jacques FIESCHI

SERIAL NO.: Not Yet Known      ART UNIT: Not Yet Known

FILED: Herewith      EXAMINER: Not Yet Known

TITLE: METHOD AND SYSTEM FOR GENERATING AND USING A VIRUS FREE  
FILE CERTIFICATE

ATTORNEY DOCKET NO.: 909.0028USU / FR919990117

Hon. Commissioner of Patents and Trademarks  
Washington, D.C. 20231

**PRELIMINARY AMENDMENT**

Sir:

This Preliminary Amendment is herewith filed in conjunction with the filing of a National Stage U.S. Patent Application which claims priority from European Patent Application No. 00480006.6, filed January 6, 2000.

Please amend the European Patent Application as set forth below:

**In the Specification**

On page 1, before "Technical Field", insert --This application claim foreign priority benefits under Title 35, United States Code, 119 of European Application No. 00480006.6, filed January 6, 2000, the contents of which are hereby incorporated by reference into this application.--

**In the Claims**

In Claim 2, line 1, delete "the preceding claim" and insert --claim 1--.

In Claim 3, line 1, delete "any one of the preceding claims" and insert --claim 1--.

In Claim 3, line 4, before "file", replace "the" with --a--.

In Claim 4, line 1, delete "any one of the preceding claims" and insert --claim 1--.

In Claim 8, line 1, delete “any one of the preceding claims” and insert --claim 1--.

- receiving (300) a virus-free certificate request for a file from a system (100, 101);
- determining (301) whether the file is virus-free or not;

- generating (303, 304) a virus-free certificate (200) comprising a file signature (207) for certifying that said file is declared virus-free by the virus-free certificate authority (102); sending (306) back in response to the virus-free certificate request the virus-free certificate (200).

- receiving (300) a virus-free certificate request for a file from a system (100, 101);
- determining (301) whether the file is virus-free or not;

- generating (303, 304) a virus-free certificate (200) comprising a file signature (207) for certifying that said file is declared virus-free by the virus-free certificate authority (102); sending (306) back in response to the virus-free certificate request the virus-free certificate (200).

2

In Claim 13, line 1, delete “any one of claims 11 to 12” and insert –claim 11--.

In Claim 14, line 1, delete “any one of claims 11 to 13” and insert –claim 13--.

In Claim 15, line 1, delete “any one of claims 11 to 14” and insert –claim 11--.

16. (Amended) A system[, preferably a server (101) or client (100) system, for carrying out the method according to any one of claims 11 to 15.] having a processor that determines that a file is virus-free, said processor executing a program for implementing a method having steps of:

- determining (400) whether a virus-free certificate (200) is associated with a file;

if a virus-free certificate is associated with the file:

- authenticating (404) the virus-free certificate (200), said virus-free certificate comprising a certificate signature (206);

authenticating (407) the file, said virus-free certificate (200) comprising a file signature (207), said file signature certifying that said file has been declared virus-free by a virus-free certificate authority (102).

17. (Amended) A computer program recorded on a computer-readable medium comprising instructions for [carrying out the] executing a method [according to any one of claims 11 to 15.] having steps of:

- determining (400) whether a virus-free certificate (200) is associated with a file;

if a virus-free certificate is associated with the file:

- authenticating (404) the virus-free certificate (200), said virus-free certificate comprising a certificate signature (206);

authenticating (407) the file, said virus-free certificate (200) comprising a file signature (207), said file signature certifying that said file has been declared virus-free by a virus-free certificate authority (102).

### **In the Abstract**

Please delete lines 1-25 and insert

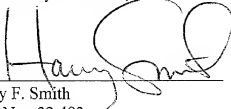
-- **Abstract of the Disclosure**

A method and system are disclosed for generating and using a virus-free file certificate. The method, for use in a virus-free certificate authority (102), includes the steps of: receiving (300) a virus-free certificate request for a file from a server (101) or a client (100) system; determining (301) whether the file is virus-free or not; if the file is declared virus-free by the virus-free certificate authority (102): generating (303, 304) a virus-free certificate (200) including a file signature (207) for certifying that said file is declared virus-free by the virus-free certificate authority (102); sending (306) back in response to the virus-free certificate request the virus-free certificate (200). The method, for use in a server (101) or client (100) system, includes the steps of: determining (400) whether a virus-free certificate (200) is associated with a file; if a virus-free certificate is associated with the file: authenticating (404) the virus-free certificate (200), said virus-free certificate including a certificate signature (206); authenticating (407) the file, said virus-free certificate (200) including a file signature (207), said file signature certifying that said file has been declared virus-free by a virus-free certificate authority (102).--

#### REMARKS

The foregoing amendment removes the multiple dependencies from the European Application. The foregoing amendment also rewrites certain dependent claims into independent form. A favorable consideration that results in the early allowance of pending claims 1-17 is earnestly solicited.

Respectfully submitted,



Harry F. Smith  
Reg. No.: 32,493  
Attorney for Applicants

9/19/2000  
Date

OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.  
One Landmark Square  
Stamford, CT 06901

Telephone: (203) 327-4500  
Facsimile: (203) 327-6401

## METHOD AND SYSTEM FOR GENERATING AND USING A VIRUS FREE FILE CERTIFICATE

### *Technical field*

The present invention relates to computer virus and more particularly to a method and system for generating and using a virus-free file certificate.

### *Background of the invention*

Among all computing and networking security issues, the most important cause of concern does not come from intrusions, but from the widespread proliferation of viruses. Viral infections represent the great majority of all security incidents.

### **Virus Protection**

Virus protection for large organizations has become more and more complex and difficult because of :

- the combined use of heterogeneous systems and practices,
- the widespread use of distributed or client/server systems, and
- the free exchange of data files via network sharing, e-mail, Internet ...

Until recently, viral infections threatened only data residing on storage media, such as hard drives and floppy disks. However, with the emergence of macro viruses, the threat has spread to applications. Most organizations are not aware of this level of penetration and are not organized to manage and prevent virus attacks. An effective virus protection software must prevent infections rather than simply treating them after they have already occurred. Anti-virus solutions need a uniform plan, with a centralized control, automated virus signature updates, and support for multiple platforms, protocols, and file types.

FR 9 99 117/YNE

005540-01-0000

## Computer Viruses

A computer virus is any program created to reproduce itself. A virus reproduces itself by attaching itself to programs, files, or even to boot sectors of disks. A virus is activated when the infected file or disk is opened or accessed. Once a virus resides in a memory, it can attach itself to the next file or disk accessed, and so on. A virus may be designed to do harm. A virus may also have unintended consequences by overwriting important computer information and by causing costly inconveniences to users and network managers. There are four general types of computer virus:

- **File Viruses** (including macro viruses), which are attached to files;
- **Boot sector Viruses** in which the boot sectors of floppy or hard disks are infected;
- **Master Boot Record (MBR) Viruses** which infect the disk master boot record; and
- **Multi-partite Viruses** that are a combination of a file virus and a boot sector virus.

## Virus Disguises

Viruses need to avoid detection in order to succeed in corrupting target computers. Simple viruses, with easily detectable signatures are giving way to more sophisticated virus types:

- **Polymorphic Viruses** : they change their signature, or profile, each time they are activated so that a fixed signature filter will miss them.
- **Stealth Viruses** : they attempt to hide their presence by intercepting interrupt services and by feeding back false information to anti-virus products and end users.
- **Encrypted Viruses** : they are delivered within an encrypted file and are undetectable by a simple anti-virus.

## Sources of Infection

FR 9 99 117/YNE



Every improvement in network and communication technologies opens new avenues through which viruses can infect your system. Most of former viruses were boot sector viruses, in which the boot sectors of floppy or hard disks were infected.

### **Macro Viruses**

As stated earlier, the creation of macro viruses has changed this environment dramatically. A macro virus is a set of instructions comprising powerful macro routines initially designed for word processing and spreadsheet applications. These macro languages enable a myriad of useful functions which can be imbedded into a document and which can be executed when the document is opened for view or use.

### **Internet**

With the exploding development of the Internet, viruses have catastrophic possibilities. The Internet introduces two different virus threats.

- The first threat is caused by the download of files comprising viruses when these files are browsed or transferred using for instance FTP (File Transfer Protocol) routines. Public shareware (shared software) and executable routines of all types, including formatted presentations, are a growing source of virus infection. Furthermore, new Internet virus threats are beginning to appear in the form of malicious JAVA and Active-X applets.
- The second threat comes from electronic mail (e-mail). Most Internet e-mail systems provide a very rich capability to attach formatted documents to mail sent over the network. These e-mail messages can be broadcast to individuals or groups of individuals with the simple stroke of a key! Infected documents or files can flood a corporate network through gateways and mail servers. As networking, telecommunications, remote access, message systems supporting attachments of all kinds become more and more common, viruses will exploit these new electronic pathways to attack systems that were heretofore unreachable.

### **Groupware Complications**

A third trend in networking also exacerbates the virus threat: the trend towards the deployment of Groupware applications such as Lotus Notes, Microsoft Exchange, Novell Groupwise, ...

Since the active and repeated sharing of documents over the network is at the core of these applications, they represent a fertile ground for the deployment of macro viruses. A Groupware application not only acts as a repository for shared documents, but, due to its collaborative function, it simultaneously broadcasts files to associated work groups. The broadcast of files significantly multiplies the possibility of accidentally deploying mail infected by attached macro viruses and makes Groupware protection a high priority.

### **Symptoms of Virus Infection**

Most viruses attempt to remain undetected as long as possible to extend their destructive influence. Therefore, most viruses do not produce any recognizable profile or signature that would allow to trap them by scanning the software. However, viruses perform actions that do not look like normal computer operations or user operations. These abnormal actions can be detected by intelligent anti-virus software. Fortunately, many viruses have telltale symptoms and may inadvertently give off signals that can alert users and virus protection software to their presence.

Some of these symptoms include:

- Increase in byte length of files,
- Alterations of a file's time stamp,
- Delayed program loading or activation,
- Reduced performance,
- Lower system resources, available memory, disk space,
- Bad sectors on floppies and hard drives,
- Strange or non-standard error messages,
- Non-standard screen activity, display fluctuations,

**FR 9 99 117/YNE**

- Program inoperability (failing to execute),
- Incomplete or failed system boots, and
- Uninitiated drive writes.

## Anti-virus Software Overview

### Detecting a Virus

Viruses are becoming increasingly sophisticated and, as such, can defeat simpler, single dimension software packages. To be effective, the anti-virus software must include special-purpose, distributed applications. Applications can detect viruses using five distinct methods:

- **Signature Scanning:** This method compares the content of files against a database of virus signatures. This method requires frequent updates of the database to ensure the identification of new and changing signatures.
- **Integrity Checking:** This method compares the profile of current files and disk areas against an archived snap shop of these same items. The detected differences may indicate the presence of a virus. Check summing is the most common type of integrity checking. Unfortunately, integrity checking is generally not effective against modern stealth viruses, so further detecting means are needed.
- **Heuristic Analysis:** An artificial intelligence monitors virus-like behavior, such as trapping certain interrupt services or attempting unlikely actions such as reformatting the hard disk.
- **Polymorphic Analysis:** Polymorphic viruses are difficult to detect because they constantly change their look, particularly when they are encrypted or when they use stealth techniques to hide their presence. A polymorphic analyzer will move any suspect file to a separate, protected, location in the computer and will execute it there to see if it exhibits any virus-like behavior.

- **Macro Virus Analysis:** A specifically designed anti-virus software detects macros in files and tests them before execution.

### **Archived and Compressed Files**

In addition to the support of these five types of virus analysis, an effective anti-virus system must also be able to scan archived and compressed files. Zip (or Pkzip) and Microsoft Compression are the most common tools for archiving and compressing a file. A virus can hide inside a compressed archive, and can remain dormant or unnoticed until the infected file is extracted and released into a system. The minimum for an efficient anti-virus system is to be able to scan most current types of archives to identify viruses stored within the files they contain.

### **Frequency of Database Signature Update**

Finally, the ability of a virus software to prevent virus attacks is determined by its ability to maintain an updated virus signature database. Any anti-virus software must have an associated, easily accessible Web site, or some other online source of information, where regular virus database updates can be retrieved. Products that automate this update process by using an Internet connection to regularly collect new information have a clear advantage in this regard.

### **Real Time and Scheduled Virus Scanning**

Most anti-virus software can perform a scan of a computer in order to detect and possibly treat the viruses found at that time. This process is called scanning. Scanning a computer for viruses can occur :

- at regular intervals under the control of a scheduler, or
- as an on-demand operation manually executed, or
- as an event-activated operation (usually in response to some recognizably "illegal" behavior by a potential virus).

In addition, viruses can be detected in real time, when they are received. This capability is important because if viruses can be detected when they attempt to enter within a system (computer, data repository, server ...), then it is possible to prevent them from corrupting other files. Oftentimes, a scheduled scan may occur after a virus has already entered within a computer and has corrupted other files. Obviously, the earlier a virus can be detected, the better.

To be truly useful, an anti-virus software must have the ability to perform all types of scans.

### **Certificate**

A Certificate is a structure that contains a public value (i.e. a public key) associated with an identity. For instance, within a X.509 Certificate, the public key is bound to a "user's name". A third party (a Certificate Authority) attests that the public key belongs to the user. A X.509 Certificate is a very formal structure and comprises different elements:

- **Subject:** This is the "user's name" (the Subject can be any identity value).
- **Issuer:** This is the name of the third party that has issued/generated the certificate. This third party is the Certificate Authority (CA).
- **Public Key Value:** This is the public key of a public/private key pair. An associated field defines the public key algorithm that must be used, for instance a RSA, Diffie-Hellman or DSA public key.
- **Validity:** Two fields are used to define the period of validity (valid from date 1 and valid to date 2).
- **Serial Number:** This field provides a unique Certificate serial number for the issuer.

- **Signature:** The signature is an encrypted digest generated by the Certificate Authority (CA) for authenticating the whole certificate. The digest results from the hashing of the Certificate. The digest is encrypted using the CA private key. The encrypted digest which is the signature, "certifies" that the Subject is the "owner" of the public and private keys.

### Certificate Verification

The Certificate needs to be verified to ensure that it is valid. This is a quite complex process. The verification by an end user of a Certificate comprises the checking of the following elements:

- Valid (or any) Subject and Issuer names are defined in the Certificate.
- The Certificate is not expired (checking of the Validity period field).
- The Certificate has not been revoked (this may be determined by obtaining a current Certificate Revocation List from the CA).
- The signature on the Certificate is valid (the signature is not verified by using the Certificate's public key but by using the CA public key).

The method for validating the signature is quite simple, and comprises the steps of:

- extracting the issuer's name (CA name) from the Certificate;
- locating the issuer's Certificate (CA Certificate) or the issuer's public key (CA public key).
- checking that the end user's Certificate signature was generated by the issuer (CA) using the issuer's public key (CA public key).

Certificates are generated by a Certificate Authority (CA). Two main methods can be used:

**FR 9 99 117/YNE**

- **Distributed Generation:** The private/public key pair is generated by the end user. The end user requests the CA to build a Certificate including the end user public key. The public key is then sent to the CA for certification. If the request is valid then the CA returns a Certificate associating the user identity with the user public key to the end user.

Of course these two methods can be combined in any system, because trusted CA keys are generated by the Certificate Authority (CA).

Current anti-virus method are becoming more and more complex due to:

- the number of viruses,
- the difficulty to find them, and
- the fact that their signature can change with time or environment.

Virus are coming from everywhere and especially from the Internet network. The time required to check a disk within a computer system, becomes more and more important. Furthermore, the checking of a disk involves the use of resources which may prevent the normal use of the computer system.

An object of the present invention is to improve current anti-virus methods and to provide a new method using file Certificates similar to X.509 Certificates used to

authenticate an identity. A specific process associates a Certificate with a file to speed up and improve the anti-virus processing.

It is another object of the present invention to associate files with a Certificate in view of simplifying the anti-virus processing of said files.

It is another object of the present invention to validate a file against all known viruses. A Certificate is added to the file. The Certificate includes a signature made by a trusted server. This signature avoids local computer systems to check this file for all existing virus. The trusted server validates the file against all known viruses. This server can use one or several anti-virus checkers. In case of new virus only the certificates are changed or updated. The only process performed by the local computer system is to verify the file against the signature included in the Certificate.

It is another object of the present invention to drastically simplify the computing resources used for virus detection. Files on Web Servers are downloaded with their certificates suppressing the risk of virus. The full anti-virus is done once instead of being done locally on each computer system.

It is another object of the present invention to generate a virus-free Certificate associated with a file using a trusted Anti-virus Certificate Authentication Server.

It is another object of the present invention to use this virus-free Certificate on a workstation to perform an anti-virus detection.

### ***Summary of the invention***

The present invention relates to computer virus and more particularly to a method and system for generating and using a virus-free file certificate. The method, for use in a virus-free certificate authority, comprises the steps of:

**FR 9 99 117/YNE**



- receiving a virus-free certificate request for a file from a server or a client system;
- determining whether the file is virus-free or not;

if the file is declared virus-free by the virus-free certificate authority:

- generating a virus-free certificate comprising a file signature for certifying that said file is declared virus-free by the virus-free certificate authority;
- sending back in response to the virus-free certificate request the virus-free certificate.

The method, for use in a server or client system, comprises the steps of:

- determining whether a virus-free certificate is associated with a file;

if a virus-free certificate is associated with the file:

- authenticating the virus-free certificate, said virus-free certificate comprising a certificate signature;
- authenticating the file, said virus-free certificate comprising a file signature, said file signature certifying that said file has been declared virus-free by a virus-free certificate authority.

### ***Brief description of the drawings***

The novel and inventive features believed characteristics of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative detailed embodiment when read in conjunction with the accompanying drawings, wherein :

**FR 9 99 117/YNE**

- Figure 1 describes the different entities involved in the anti-virus system according to the present invention.
- Figure 2 describes the content of a virus-free Certificate according to the present invention.
- Figure 3 is a flow chart of the method of requesting and generating a virus-free Certificate for a file according to the present invention.
- Figures 4a and 4b are a flow charts of the method of using a virus-free Certificate in a workstation according to the present invention.

### ***Preferred embodiment of the invention***

#### **Introduction**

Figure 1 describes the different entities involved in the anti-virus system disclosed in the present invention. In most of the cases, the file that the Client Workstation (100) requires, is stored in a Web / File Server (101). A Certificate, stored in a directory within the Web / File Server (101), is associated with this file. The Certificate is provided by a Virus-Free Certificate Authority Server (102) after request. Said request is sent by the Web / File Server (101) to the Virus-Free Certificate Authority Server (102) through a LAN / WAN (Local Area Network / Wide Area Network) (103) which can include the Internet network. The Client Workstation (100) then downloads both file and associated Certificate in a directory and asks its anti-virus program to check the file. This checking process does not use any standard anti-virus program but is based on the previously downloaded Certificate. The only verification required for determining whether the file is virus-free or not, is the verification of the signature comprised in the Certificate. All the above mentioned method will be better understood with respect to Figures 2, 3 and 4a/4b.

#### **Virus-free Certificate**

FR 9 99 117/YNE

- an anti-virus name and level;
- a signature of the file.

- **File name (201):** This is the “name” of the file protected that the virus-free Certificate protects.

• **Public Key Value (203):** This is the public key of a public/private key pair. An associated field defines the public key algorithm that must be used to check the file signature, for instance a RSA , Diffie-Hellman or DSA public key. The public key is provided by the Virus-free Certificate Authority which uses the corresponding private key to build the signature of files. So the same private/public key pair may be used to build several virus-free Certificates from the same issuer. This public key within the virus-free Certificate is preferably used instead of the Virus-free Certificate Authority public key which is used to validate only the present certificate signature and not the file signature. A public key for decrypting the imbedded signature is added within the virus-free Certificate because the Virus-free Certificate Authority public key is generally longer and more complex. The validity of keys may also differ between the Virus-free Certificate Authority public key and the virus-free Certificate public key. Anyway, because the virus-free Certificate is signed by the Virus-free Certificate Authority, the use of the virus-free Certificate public key is secure.

- **Validity (204):** Two fields are used to define the period of validity (valid from date 1 and valid to date 2).
- **Serial Number (205):** This field provides a unique virus-free Certificate serial number for the issuer.
- **Certificate Signature (206):** The certificate signature is an encrypted digest generated by the Virus-free Certificate Authority (VCA) for authenticating the whole Certificate. The digest results from the hashing of the virus-free Certificate. The digest is encrypted using the VCA private key. The certificate signature results from the encrypted digest and “certifies” that the file signature is encrypted by the private key associated with the virus-free certificate public key (203). The Virus-free Certificate Authority (VCA) public key is different from the virus-free Certificate public key and is either preloaded in the web browser or given by a trusted entity. The VCA public key is used to retrieve the original hashing of the full certificate. The Virus-free Certificate Authority (VCA) can use the same set of virus-free certificate private / public keys (203) for all the files generated during a given period of time so the cross-checking of the issuer authentication can be easily performed time to time, when a new set of keys is used. Once the virus-free Certificate public key for a issuer is validated, it can be reused for several files certified by the same issuer which reduces the number of virus-free Certificate public keys.
- **File Signature (207):** The File Signature is verified using the public key value given in the virus-free Certificate
- **Anti-virus Checker (208):** This field gives an indication of how the virus-free Certificate has verified that the file was virus-free. The Anti-virus Checker comprises the

00655010\*091900

- 00655010\*091900

00655010\*091900

00655010\*091900

- 00655010\*091900

If a virus is detected,

- (307) the VCA Server answers the requester with an information concerning the detected virus.
- (308) Eventually, the VCA Server sends back to the requested a corrected file.

If no virus is detected,

- (303) A signature of the file is established.
- (304) The virus-free Certificate is prepared with this signature.
- (305) The virus-free Certificate and the requester identification, are then stored in the VCA. The requester identification may include the file location within the Web / File Server or workstation. The file location is useful when the requester needs a regular and automatic update of the virus-free Certificate (for instance, when the virus-free Certificate expires or when a new level of anti-virus program is provided). In that case the VCA can access the file and can update the virus-free Certificate without any action from the Server or workstation.
- (306) Finally, the virus-free Certificate is sent to the requester.

For a better understanding, the VCA is shown in the present embodiment as an independent Server. However, the VCA can be located within a Web / file Server (101). It is possible for a master Certificate Authority server to delegate virus-free Certificate establishment to trusted servers or workstations.

### **Virus-Free Certificate Utilization**

Figures 4a and 4b describe the process of using the virus-free Certificate in a workstation according to the present invention. A File is downloaded with its free-virus  
**FR 9 99 117/YNE**

Certificate onto a workstation. The anti-virus program performs a checking on this incoming file. The anti-virus program can also check all files assigned by configuration. Some files may have an associated virus-free Certificate, other files may have no Certificate. The present method of a using a free-virus Certificate in a workstation comprises the following steps:

- (400) When scanning files, the anti-virus program first looks for the virus-free Certificate associated with the file to check. The virus-free Certificate may be in the same directory as the file or in a specific directory with all free-virus Certificates. Other settings may be defined but the two settings above are the these used in the present embodiment.

- (401) The virus-free program looks for the virus-free Certificate:

If a virus-free Certificate for this file is not found:

- (402) The process goes on as described in the background art.
- (403) The process goes on by checking the next file.

If the virus-free Certificate for this file is found:

- (404) The virus-free Certificate is authenticated using the certificate signature and the VCA public key. The VCA public key is in the workstation or if not must be retrieved through a secure channel. The VCA server may be authenticated by another CA having the required public key.
- (405) Date of expiration, issuer name (VCA name), in addition to the certificate signature determined in the previous step (404) are checked and validated. The anti-virus program may also be checked and in particular levels used to build the Certificate which may or may not be accepted by the local anti-virus program depending on predetermined rules.

**FR 9 99 117/YNE**

09665040-091900

- If the file signature is OK,

- If the file signature is not OK,

- (410) A log error is performed.
- (411) The normal anti-virus program is activated to check this suspect file.
- (412) Finally, the next file is checked.

The Log Error file is processed at the end of the file checking and may ask the VCA to check the file again in order to produce another virus-free Certificate. If a virus is found on this file, the security administrator will retrieve all available information to understand where the virus was introduced, who introduced it ...

### Advantages



- The proposed invention provides a better and faster way for checking files against viruses.
- Servers offer a better security for all the files they send to their clients.
- The full anti-virus checking is performed once on the virus-free Certificate Authority (VCA) Server.
- A Certificate update method is provided.
- Normal ant-virus processes may be used as backup and may handle files without Certificate. The compatibility with existing anti-virus programs is easy because a software supervisor can
- verify virus-free Certificates for files having one, and
- rely on and call any anti-virus program to perform a state of the art virus detection for other files.
- The present invention is
- for users, in line with current security strategies based on Certificate Authority and Certificates, and
- for files, an extension that can be easily deployed.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood that various changes in form and detail may be made therein without departing from the spirit, and scope of the invention.

**Claims**

1. A method, for use in a virus-free certificate authority (102), of generating a virus-free certificate (200) certifying that a file is virus-free comprising the steps of:

- receiving (300) a virus-free certificate request for a file from a system (100, 101);
- determining (301) whether the file is virus-free or not;

if the file is declared virus-free by the virus-free certificate authority (102):

- generating (303, 304) a virus-free certificate (200) comprising a file signature (207) for certifying that said file is declared virus-free by the virus-free certificate authority (102);
- sending (306) back in response to the virus-free certificate request the virus-free certificate (200).

2. The method according to the preceding claim wherein the virus-free certificate request comprises:

- a list of one or a plurality of anti-virus programs to execute on the file to determine whether the file is virus-free or not.

3. The method according to any one of the preceding claims wherein the virus-free certificate request comprises:

- the file for which a virus-free certificate is requested.

4. The method according to any one of the preceding claims wherein said step of determining whether the file is virus-free or not comprises the further step of:

- executing (301) one or a plurality of anti-virus programs on said file for detecting viruses.

5. The method according to any one of the preceding claims wherein the virus-free certificate further comprises:

- the list of the one or plurality of anti-virus programs (208) that have been executed on the file.

6. The method according to any one of the preceding claims wherein the virus-free certificate (200) further comprises:

- a file identification (201);
- a virus-free certificate authority identification (202);
- a public key (203) for decrypting the file signature;
- a certificate signature (206) for authenticating the virus-free certificate;
- a indication of the virus-free certificate validity (204).

7. The method according to any one of the preceding claims comprising the further steps (305) of:

- identifying the system (100, 101) where the file and associated virus-free certificate are stored;
- downloading updates of the virus-free certificate.

8. The method according to any one of the preceding claims wherein the step of generating (303, 304) a file signature (207) comprises the further steps of:

- hashing the file to generate a file digest;
- encrypting the file digest using a private key.

**FR 9 99 117/YNE**

9. A system, preferably a virus-free certificate authority (102), for carrying out the method according to any one of the preceding claims.

10. A computer program comprising instructions for carrying out the method according to any one of claims 1 to 8.

11. A method, for use in a server (101) or client (100) system, of determining that a file is virus-free comprising the steps of:

- determining (400) whether a virus-free certificate (200) is associated with a file;

if a virus-free certificate is associated with the file:

- authenticating (404) the virus-free certificate (200), said virus-free certificate comprising a certificate signature (206);
- authenticating (407) the file, said virus-free certificate (200) comprising a file signature (207), said file signature certifying that said file has been declared virus-free by a virus-free certificate authority (102).

12. The method according to the preceding claim wherein said step of authenticating (407) the file comprises the further steps of:

- decrypting the file signature (207) using a public key (203) comprised in the virus-free certificate (200).
- hashing the file to generate a file digest;
- comparing the decrypted file signature with the generated file digest.

13. The method according to any one of claims 11 to 12 wherein the step of authenticating the virus-free certificate comprises the further step of:

**FR 9 99 117/YNE**

- 0665041-091900

**Abstract**

The present invention relates to computer virus and more particularly to a method and system for generating and using a virus-free file certificate. The method, for use in a virus-free certificate authority (102), comprises the steps of:

- receiving (300) a virus-free certificate request for a file from a server (101) or a client (100) system;
  - determining (301) whether the file is virus-free or not;
- if the file is declared virus-free by the virus-free certificate authority (102):
- generating (303, 304) a virus-free certificate (200) comprising a file signature (207) for certifying that said file is declared virus-free by the virus-free certificate authority (102);
  - sending (306) back in response to the virus-free certificate request the virus-free certificate (200).

The method, for use in a server (101) or client (100) system, comprises the steps of:

- determining (400) whether a virus-free certificate (200) is associated with a file;
- if a virus-free certificate is associated with the file:
- authenticating (404) the virus-free certificate (200), said virus-free certificate comprising a certificate signature (206);
  - authenticating (407) the file, said virus-free certificate (200) comprising a file signature (207), said file signature certifying that said file has been declared virus-free by a virus-free certificate authority (102).

Figure 1

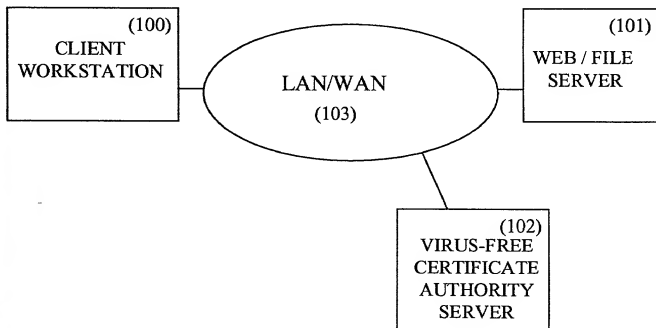


FIG. 1

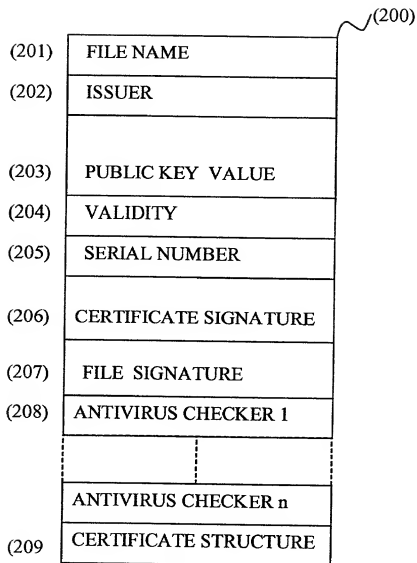


FIG. 2



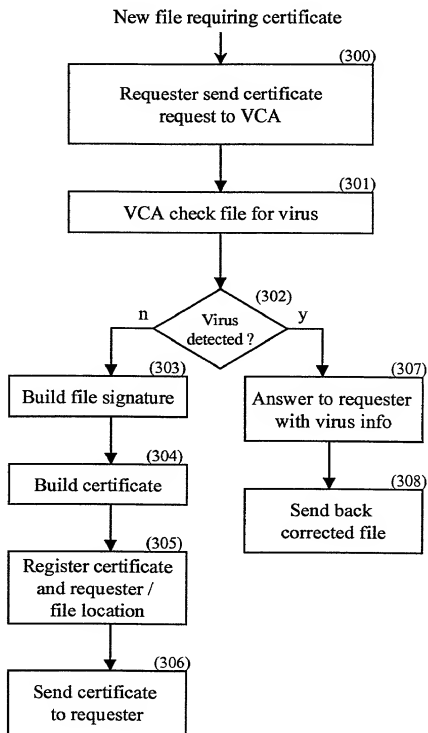


FIG.3

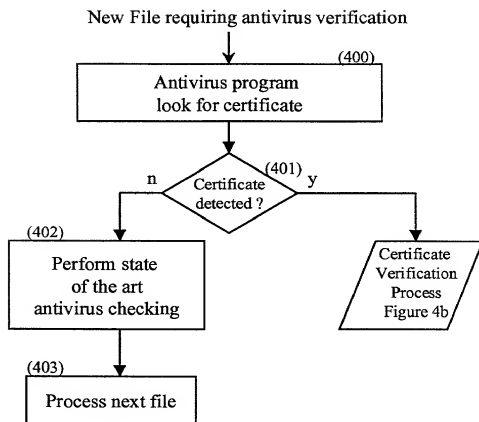


FIG. 4a

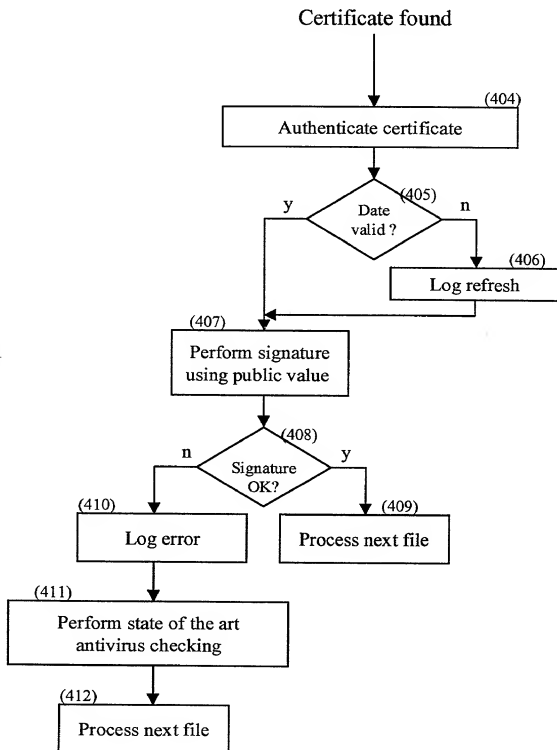


FIG. 4b

**DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION**Docket No. **909.0028USU / FR919990117**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**METHOD AND SYSTEM FOR GENERATING  
AND USING A VIRUS FREE FILE CERTIFICATE**

the specification of which

(check one)   X   is attached hereto.           was filed on as Application Serial No.

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, 119 of any foreign application(s) for patent or inventor's certificate(s) listed below and have also identified below any foreign application(s) for patent or inventor's certificate(s) having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

Priority Claimed

00480006.6  
(Number)Europe  
(Country)06 January 2000  
(Day/Month/Year Filed)  X   Yes        No

I hereby claim the benefit under Title 35, United States Code, 119(e) and 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Prov. Appl. No.)

(Filing Date)

(Status)

(patent, pend., abandon.)

**POWER OF ATTORNEY:** As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith, as follows:

<b>NAMES</b>	<b>REGISTRATION NUMBERS</b>
Manny W. Schechter	Reg. No. 31,722
Christopher A. Hughes	Reg. No. 26,914
Edward A. Pennington	Reg. No. 32,588
John E. Hoel	Reg. No. 26,279
Joseph C. Redmond, Jr.	Reg. No. 18,753
Paul J. Otterstedt	Reg. No. 37,411
Douglas W. Cameron	Reg. No. 31,596
Wayne L. Ellenbogen	Reg. No. 43,602
Stephen C. Kaufman	Reg. No. 29,551
Daniel P. Morris	Reg. No. 32,053
Louis J. Percello	Reg. No. 33,206
Robert M. Trepp	Reg. No. 25,933
David M. Shofi	Reg. No. 39,835
Louis P. Herzberg	Reg. No. 41,500
Lauren C. Bruzzone	Reg. No. 35,082
Richard M. Ludwin	Reg. No. 33,010
Marc A. Ehrlich	Reg. No. 39,966
Marian Uderweiser	Reg. No. 46,134

**SEND CORRESPONDENCE TO:**

Harry F. Smith, Esq.  
OHLANDT, GREELEY, RUGGIERO & PERLE, L.L.P.  
One Landmark Square  
Stamford, CT 06901

**DIRECT TELEPHONE CALLS TO:**

Harry F. Smith, Esq.  
Telephone:(203)327-4500  
Facsimile:(203)327-6401

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

FULL NAME OF INVENTOR **LE PENNEC** FIRST NAME **JEAN-FRANÇOIS** MIDDLE INITIAL  
 RESIDENCE & CITY **11 CHEMIN DE LA SERENA** STATE OR COUNTRY **FRENCH** CITIZENSHIP  
 POST OFFICE ADDRESS **06100 NICE, FRANCE** P.O. ADDRESS CITY & STATE ZIP CODE  
**SAME**

Inventor's  
Signature

*Jean François Le Penne* Date *Sept 16<sup>th</sup>, 2000*

FULL NAME OF INVENTOR **HERICOURT OLIVIER** MIDDLE INITIAL  
 RESIDENCE & CITY **RESIDENCE LE FLORE** STATE OR COUNTRY **FRENCH** CITIZENSHIP  
**62 CHEMIN DU VAL FLEURI**  
**06800 CAGNES SUR MER, FRANCE**  
 POST OFFICE ADDRESS **SAME** P.O. ADDRESS CITY & STATE ZIP CODE

Inventor's  
Signature

*Olivier Hericourt* Date *Sept 11, 2000*

FULL NAME OF INVENTOR **FIESCHI JACQUES** MIDDLE INITIAL  
 RESIDENCE & CITY **LE ROSE GARDEN - BAT B** STATE OR COUNTRY **FRENCH** CITIZENSHIP  
**60 AVENUE P. CEZANNE**  
**06700 SAINT LAURET DU VAR, FRANCE**  
 POST OFFICE ADDRESS **SAME** P.O. ADDRESS CITY & STATE ZIP CODE

Inventor's  
Signature

*Jacques Fieschi* Date *Sept 12, 2000*